RESEARCH CENTERS
Applications
Careers
Convergence
Data Center
LANs
Net/Systems Mgmt.
NOSes
Outsourcing
Routers/Switches
Security
Service Providers
Small/Med.Business
Storage
WAN Services
Web/e-commerce
Wireless/Mobile

SITE RESOURCES
Daily News
Newsletters
This Week in NW
Tests/Reviews
Buyer's Guides
Opinion
Forums
Special Issues
How to/Primers
Case Studies
Network Life
Encyclopedia
IT Briefings
XML

Layer 8
- Yet another Google announcement
- Chiropractors love laptops
- More

NETWORKING FOR SMALL BUSINESS

TODAY'S NEWS
- Microsoft to unveil a capacity mgmt. tool as part of broader mgmt. suite
- Cisco warns of ICMP-based attacks on routers
- HSBC warns 180,000 over retailer's security breach

Public-key encryption for dummies

By MIKE ROTHMAN
Network World, 05/17/99

Printer friendly version
Send this article to a colleague

As the world increasingly turns to electronic business, electronic credentials that prove identity are becoming a critical necessity. Much like a passport proves identity in the offline world, public-key infrastructure (PKI) delivers a way to prove identity in the online world.

PKI is fast becoming the cornerstone of information security technology for a large number of companies.

PKI ensures that people are who they say they are and also proves that documents haven't been tampered with, which is critical when conducting online transactions, such as placing orders or transferring money. Here's a simplified look at these state-of-the-art passports to the online world.

The magic of PKI occurs through the use of extremely long prime numbers, called keys. Two keys are involved - a private key, which only you have access to, and a public key, which can be accessed by anyone. The two keys work together, so a message scrambled with the private key can only be unscrambled with the public key and vice versa. The more digits in these keys, the more secure the process.

Just as you prove your identity through a handwritten signature offline, you use a digital signature to prove your identity online. But without seeing a person sign the document, how can you prove it's the right person?

This is where public-key cryptography comes in. A large piece of

- Cisco to acquire Topspin for $250 million
- IBM shows Q1 growth but falls short of forecasts
- More breaking news

data set to be encoded - for instance, a document - is run through a complicated mathematical computation to generate a single large number, called a hash. The original data and the hash are inextricably linked. If either changes, the hash won't match and the message cannot be decoded.

To digitally sign a document, a hash is taken of the document and then signed with a user's (let's call him Bob) private key. Data scrambled with Bob's private key can only be unscrambled with Bob's public key. Any entity can verify the validity of the document by unscrambling the hash with Bob's public key and checking that against another hash computed from the received data.

If the hashes match, the data was not tampered with and Bob's digital signature is on it. But because I didn't watch Bob sign the document, I don't know that it wasn't signed by an imposter. This issue is solved because only Bob has his private key, and so he is the only one who could have signed the document.

How do I know I have the correct key to verify the signature? This is where the concept of trust enters the system, creating the need for a certificate authority to verify online identity.

The certificate authority is like an online passport bureau - a trusted entity that makes the PKI system work. The private key is securely generated by Bob, and after verifying Bob's identity, the certificate authority signs Bob's public key with its own private root key. The combination of Bob's public key and the signature of the certificate authority completes Bob's digital certificate.Bob's digital certificate is his online passport, validated by the certificate authority's watermark.

Let's look at how all this works together in a simple transaction. Bob wants to send Alice a confidential e-mail. Bob would use Alice's public key, stored in her certificate, to scramble the message. When Alice receives the message, she uses her private key to unscramble it. Because no one else possess Alice's private key, only she can unscramble the message.

The process is similar in complex transactions. Let's say Bob wants to let Alice order products from his Web site. When Alice is ready to buy, Bob requests that she prove her identity. Alice signs the order with her private key, which was issued by a certificate authority we'll call TrustCo. She then sends the package consisting of the order and the digital signature to Bob.

Bob needs to get Alice's and TrustCo's digital certificate to verify the signature. He validates Alice's certificate by verifying TrustCo's signature (remember TrustCo signs Alice's public key, thus forming the certificate), and then uses Alice's certificate to validate the signature on the order. If all those tests pass, Alice is actually Alice.

Like any security technology, digital signatures used in the PKI model aren't perfect. If the certificate authority's root key is stolen, then anyone can create digital certificates, which compromises the trust level of the certificate authority and makes all the certificates from that certificate authority null and void. Certificate authorities go to great lengths to keep their keys secure, including armored bunkers. Additionally, if Bob loses his private key, or if it's stolen, then anyone possessing the private key can pose as Bob.

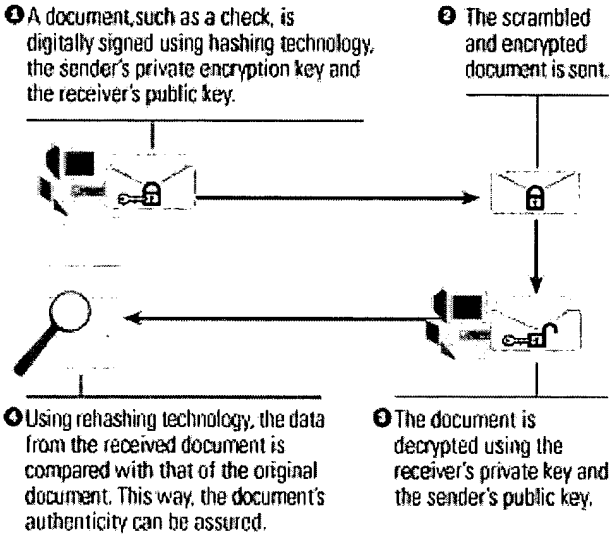
More importantly, thousands of applications used throughout businesses need to be PKI-ready. Applications need to know how to ask Bob to sign data and how to validate the data using certificates. For PKI to become a widely used technology, it must become a transparent part of everyday software, so end users don't need to understand all the complexity behind keys, hashes and digital certificates.

Rothman is executive vice president of SHYM Technology, a software company that makes PKI wares. He can be reached at mrothman@shym.com or www.shym.com.

HOW IT WORKS

Public-key infrastructure

Electronic business is picking up, and with it the need for secure electronic credentials is increasing. PKI is a way to prove identity in the online world. It also certifies that documents have not been tampered with.



Feedback

Tell us your thoughts on this article or the issues raised in it. We'll cc: the author and editors on all comments.

Comments:

Name:

E-mail address:

Can we post your comments in an online forum on the topic?

○ **Yes** ○ **No**

What did you think of this article?

○ **Very useful** ○ **Somewhat useful** ○ **Not at all useful**

Would you want to see:

- ○ **More articles on this topic**
- ○ **Fewer articles on this topic**

Submit

Thank you! When you click Submit, you'll be taken back to this article.

RELATED LINKS

Feedback

Tell us your thoughts on this article or the issues it raises.

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	28518	authentication and keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:40
L2	12459	authentication with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:17
L3	135	authentication with matching with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:17
L4	1347	(valida$6 verif$6 authentication) with (comapr$4 match$4) with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:33
L5	1038	4 and first and second and (trustee server third) and (valida$6 verif$6 authentication) with (comapr$4 match$4) with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:34
L6	210	5 and "705"/$.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:35
L7	106	6 not @py>"2002"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:35
L8	73	7 and public and private	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:21

L9	596	authentication with (system server) and matching with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:58
L10	120	9 and transmitting with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:55
L11	111	9 and "705"/$.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:57
L12	14	10 and 11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:57
L13	1545	authentication with (system server) and transmitting with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 12:59
L14	6	authentication with (system server) and transmitting with matching with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:00
L15	1242	authentication and transmitting near3 key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:01
L16	176	15 and "705"/$.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:03

L17	8	("6300873" "5870723" "5761309" "4500750").pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:12
L18	1	17 and 16	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:03
L19	6	authentication and transmitting with matching with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:17
L20	6	authentication and transmitting with matching with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:19
L21	2458	authentication and transmitting with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:19
L22	79	transmitting with matching with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:21
L23	0	22 and encypt$ and transmitting with matching with key	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:20
L24	35327	encryption and keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:22

L25	855531	encryption and matching keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:22
L26	1176	encryption and matching with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:22
L27	138	26 and encryption and (transmit$6 send$4) with matching with keys	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:25
L28	117	27 and (verif$7 authentica$7 validat$6)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/04/15 13:27